Doc Code: AP.PRE.REQ

OIPE IAP97 DEC 21 2006 PATENT & TRADEMARK OFFICE

PTo/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW		**Docket Number (Optional)** 550-192
	Application Number 09/731,060	Filed December 7, 2000
	First Named Inventor NEVILL	
	Art Unit 2194	Examiner Zhen, Li B.

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ Applicant/Inventor

☐ Assignee of record of the entire interest. See 37 C.F.R. § 3.71. Statement under 37 C.F.R. § 3.73(b) is enclosed. (Form PTO/SB/96)

☒ Attorney or agent of record 33,149 (Reg. No.)

☐ Attorney or agent acting under 37CFR 1.34.
Registration number if acting under 37 C.F.R. § 1,34 ______

[illegible]
Signature

John R. Lastova
Typed or printed name

703-816-4025
Requester's telephone number

December 21, 2006
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.*

☒ *Total of 1 form/s are submitted.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

NEVILL et al.

Appl. No. 09/731,060

Filed: December 7, 2000

OIPE IAP87 DEC 21 2006 PATENT & TRADEMARK OFFICE

Atty. Ref.: 550-192; Confirmation No. 8029

TC/A.U. 2194

Examiner: Zhen, Li B.

For: SCHEDULING CONTROL WITHIN A SYSTEM HAVING MIXED HARDWARE AND SOFTWARE BASED INSTRUCTION EXECUTION

* * * * * * * * * * *

December 21, 2006

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

STATEMENT OF ARGUMENTS IN SUPPORT OF PRE-APPEAL BRIEF REQUEST FOR REVIEW

In this fifth office action after withdrawal of the prior final rejection, the claims stand rejected under 35 U.S.C. §103 based on the combination Evoy (5,937,193) and newly-cited Gee (6,374,286). Like the earlier four rejections, this one is also improper.

A processing system provides both hardware instruction translation and software instruction interpretation mechanisms for supporting high level program instructions, e.g., Java bytecodes. The program instructions are all supplied to the hardware translation unit (see 64 in Fig. 9) which forwards those instructions it does not itself support to the software interpretation mechanism (see step 84 in Fig. 10). By routing all program instructions through the hardware translation unit, the hardware translation unit can monitor when it is appropriate and safe to trigger a scheduling operation for controlling multi-tasking or multi-threaded operations.

Clear Error #1: Neither Nor Gee Teach Element (iv) In The Independent Claims

Neither Evoy nor Gee discloses the independent claim feature (iv) where program instructions that are not supported by the hardware based execution unit are "forwarded to said software based execution unit for execution with control being returned to said hardware based execution unit for a next program instruction to be executed." The Examiner relies solely on

Evoy and contends that "selecting the next bytecode" in col. 7, lines 8-16 of Evoy (7:8-16) discloses the quoted claim feature. Applicants disagree. Evoy teaches at 6:61-7: 21 that although the processor normally runs in native mode, when the processor requests instructions from the platform-independent partition of the system memory, the processor switches to a platform-independent mode. An object table is scanned to see if a stored native instruction corresponds to a given bytecode. But if no corresponding native instruction exists, the object table outputs an exception signal which notifies the processor that software interpretation of the bytecode may be required (7:14-16). After executing the native instruction in the platform-independent mode, the processor increments its address counter to the next instruction which has the effect of selecting the next bytecode (7:17-20).

It would be clear to the skilled person that Evoy's "selecting the next bytecode" text on which the Examiner relies only applies when the processor is in the platform-independent mode—not when the processor is in native mode. When the processor encounters an instruction it cannot handle, software interpretation of the bytecode is used, which necessarily involves the processor having access to native instructions. So it is clear that processor will be in native mode once software interpretation is invoked. At 7:27-29, Evoy teaches that return from the platform-independent mode to native mode occurs whenever the processor accesses the native partition of system memory. But when the processor is in its normal native mode, the statements regarding "selecting the next bytecode" (7:20) do not apply.

Thus, contrary to the Examiner's assertion, 7:8-16 of Evoy does not disclose the claimed feature where control is returned to the hardware based execution unit for a next program instruction to be executed. When Evoy's processor is in native mode, the address counter points to native instructions and incrementing the address counter selects the next native instruction for execution. Control is not returned to platform independent mode once software interpretation has been invoked.

The Examiner also refers to 11:6-15 which discloses that if the "source address" value does not equal the "source end address" value, then additional platform-independent instructions must be processed. Once the source address register has been processed, control returns to block 202 to process the next platform-independent instruction. This passage can be understood with reference to Figure 6. The text at 10:32-34 explains that Figure 6 relates to a translation code routine for a translation state machine. So this 11:6-15 passage relates translating and storing Java bytecodes for subsequent execution. No Java bytecodes are actually executed in the flow chart of Figure 6. Accordingly, 11:6-15 cannot disclose the claimed feature where "control is returned to the hardware based execution unit for the next program instruction to be executed."

Clear Error # 2: Combining Evoy and Gee Does Not Teach Claim Element (v)

In feature (v) of the independent claims, the hardware based execution unit includes scheduling support logic that generates "a scheduling signal for triggering a scheduling operation to be performed between program instructions for managing scheduling between threads or tasks irrespective of whether a preceding program instruction was executed by said hardware based execution unit or said software based execution unit." The Examiner concedes that Evoy only discloses scheduling in a general sense in that an address counter is incremented after the execution of an instruction, which has the effect of selecting the next bytecode. As the Examiner admits, incrementing an address counter to select the next bytecode for translation is not scheduling between threads or tasks.

Multi-tasking operating systems require processing resources to be shared between several different programs that may be simultaneously active, and multi-threaded computer programs similarly require processing resources to be shared between different active threads. The inventors recognized drawbacks with simplistic counter or timer scheduling approaches. A simple timer-based scheduling approach may suffer from the disadvantage that scheduling operations may be inappropriately triggered at points part-way through the software interpretation of a complex program instruction in a manner that could cause a loss of data integrity should an inappropriate context switch occur. A simple counter-based scheduling requires the extra overhead of exchanging counter values between hardware-executed program instructions and software-executed program instructions. Indeed, Evoy suffers from these very problems where scheduling operations may be triggered part way through interpretation of a complex non-native program instruction.

The Examiner relies on text in Gee at 21:25-60 which simply discloses a "conventional java scheduling policy" (21:30-31) where threads are scheduled to run in accordance with priority-based rules. So Gee's conventional scheduling policy is like the known scheduling policy referenced in the background on page 2, lines 8-17. In contrast, the independent claims recite (1) returning control to the hardware based execution unit for a next program instruction to be executed (as explained above) and (2) the hardware based execution unit triggering a scheduling operation between program instructions for managing scheduling between threads or tasks irrespective of whether a preceding program instruction was executed by the hardware based execution unit or the software based execution unit. A benefit of this claimed approach is that the hardware-based execution unit can keep track of the execution of instructions, and as a result, reliably generate the scheduling signal for triggering a scheduling operation irrespective of whether the preceding instructions have been executed by hardware or software.

Clear Error #3: Evoy and Gee Do Not Address or Solve the Problems Solved in This Case

Evoy and Gee do not recognize the problems to which the missing features are directed where scheduling operations may be inappropriately triggered (simple timer-based approach) or have disadvantageous overhead of counter exchange (counter-based scheduling) in processing systems having mixed hardware based execution units and software based execution units. The failure to recognize the problem being solved by the claimed invention is a telling factor points to another clear error that the required "teaching, suggestion, or motivation to combine the references" is missing. *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998). The Examiner fails to set forth objective evidence to show motivation to include Gee's conventional scheduling in the apparatus of Evoy. Moreover, the Examiner's proposed combination does not solve the problems with permitting scheduling regardless of whether the preceding program instruction was executed by the hardware based execution unit or the software based execution unit.

In contrast, the combination of claim features (iv) and (v) prevents these problems and maintains synchronization between hardware and software with respect to a scheduling signal by sending program instructions for execution to the hardware based execution unit and returning control to the hardware based execution unit even if the program instructions to be executed have been forwarded to the software based execution unit for execution. Routing the program instructions for execution to the hardware based execution unit and returning control to the hardware based execution unit enables the scheduling operation to be triggered irrespective of whether a preceding program instruction was executed by the hardware based execution unit or the software based execution unit.

Clear Error #4: Dependent Claim Features Are Not Taught

The Examiner contends that Evoy 7:17-27 discloses the subject-matter of claim 2. But address counter (or program counter) which is completely different from the claimed counter included in the scheduling support logic. Evoy's address counter is a pointer to a memory location holding the current instruction, which is incremented after execution of each instruction. This address counter does not count up or down towards any specific value. It would not be possible to use Evoy's address counter as scheduling support logic because it could not be used to program a scheduling signal to occur when the counter reaches a particular value.

The Examiner contends that the "END reserved code" of 10:65-11:6 of Evoy corresponds to the subject-matter of claim 3 where the counter triggers generation of the scheduling signal when the predetermined count value is reached. The Examiner equates the END reserved code of Evoy with the predetermined count value. But this passage does not teach generating a scheduling signal "for managing scheduling between threads or tasks" as recited in claim 1. Instead, this passage discloses passing control to process a new instruction in the event that the new instruction is not an END reserve code. This is different from generating a scheduling signal when a counter has reached a predetermined count value as specified by claim 3.

Given the clear errors in the rejection, the rejection based on should be withdrawn and the application passed to allowance.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: [illegible]
John R. Lastova
Reg. No. 33,149

JRL:maa
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100